Multiplexed Flash Illumination for Relighting and Depth Extraction

Abstract

We multiplex flash illumination to recover both flash and ambient light information as well as extract depth information in a single exposure. Traditional photographic flashes illuminate the scene with a spatially-constant light beam. By adding a mask and optics to a flash, we can project a spatially varying illumination onto the scene which allows us to spatially multiplex the flash and ambient illuminations onto the imager. We apply flash multiplexing to enable single exposure flash/no-flash image fusion, in particular, performing flash/no-flash relighting on dynamic scenes with moving objects. We exploit the defocus of the multiplexing pattern to also infer depth information.

1 Introduction

Taking good photographs in low-light situations is challenging, and a flash is often the most practical option for very dark scenes. Unfortunately, a flash can often ruin the natural ambiance of the available lighting, producing harsh, unflattering pictures. Flash/no-flash methods [ED04; PSA*04] combine two images of a scene, one taken with a flash and one taken without, to produce a new image with the best properties of both images. While these methods work well for static scenes, the requirement of multiple exposures is a significant barrier to the average user, and infeasible for moving scenes because of the need for multiple exposures.

We propose a method for simultaneously capturing flash and ambient lighting information in a single exposure. We use a coded flash to project a high-frequency pattern onto the scene, which spatially multiplexes flash and no-flash information (see Figure 1). Spatially multiplexing flash and no-flash gives information about both the detail and color in the flash regions and the ambient illumination in the no-flash regions, though with a reduced resolution and contribution from indirect illumination due to the flash.

We build on the idea of assorted pixels [NN02; NM00] but extend it to computational illumination. We aim to spatially multiplex flash information into a single image. In contrast to previous work on temporal multiplexing of illumination, e.g. [DHT*00; WGT*05; MS05; NKGR06; SNB07], our goal is to *simultaneously* record both types of information,. Simultaneous capture is important for dynamic scenes to avoid a temporal mismatch between the images corresponding to the two lighting conditions.

Furthermore, we want to leverage the defocus information from the multiplexing light pattern in order to infer depth information. However, in contrast to previous work, [MNBN07] we seek to do so in the presence of ambient illumination and with a light pattern that is not co-axial with the lens, in order to increase light efficiency.

The main contributions of this paper are:

- The introduction of assorted flash pixels to record spatially multiplexed flash and ambient information.
- Estimation of a sparse depth map from flash defocus.
- Single exposure flash/no-flash applied to dynamic scenes.

2 Related Work

Assorted Pixels, proposed by Nayar and Narasimhan[NN02], introduced a method for sampling multiple dimensions of imaging (e.g. brightness, color spectrum, time, polarization) by mosaicing pixels that sample different dimensions into a single array of pixels. We extend this concept by allowing the illumnation to be mosaiced. Unlike traditional Assorted Pixels, in which the multiplexing occurs purely on the image sensor, we multiplex at the illumination source and must identify which pixels on the sensor are sampling along which dimension.

Figure 1: *Top: A scene photographed with and without flash. Bottom: Close-ups of two samplings of flash and no-flash pixels using our multiplexed flash illumination.*

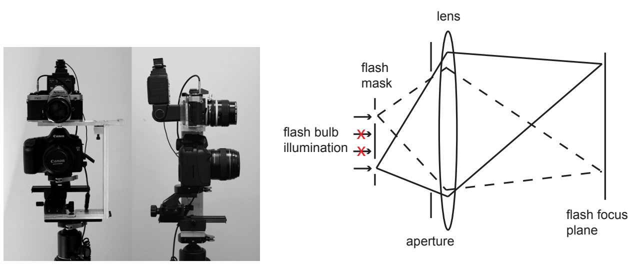
Structured lighting has been used to accomplish a variety of tasks, including depth and shape estimation[ZN06], refocusing [LCV*04; MNBN07], light transport estimation [SCG*], and direct and indirect lighting separation [NKGR06]. Many of these techniques are restricted to static scenes because they require multiple images of the scene, while our goal is to capture flash and ambient information for a scene in a single exposure. Additionally, some methods (e.g. [MNBN07]), require a coaxial camera and projector which is accomplished using a beam-splitter. Beam-splitters lose lights, and introduce glare, which is undesirable for low-light photography, our main application.

A number of approaches seek to capture a full basis of possible illumination to enable arbitrary relighting of a scene, e.g. [DHT*00; WGT*05]. This requires a large number of images to encode the full set of possible direction and, in the case of dynamic scenes, careful correction must be applied to warp the data [WGT*05]. In contrast, we seek a simultaneous capture but restrict ourselves to two illumination conditions.

Nayar et. al. [NKGR06] describe a method for fast separation of the direct and indirect component of a scene illuminated by a single light source. This method uses a sequence of high-frequency patterns projectected onto the scene to perform the separation. They also describe a single exposure version which can produce separations, albeit with a loss in resolution. We assume the scene is lit by two sources, our multiplexed flash and an ambient light source. Our goal is to separate the image into flash and ambient components by spatially multiplexing each component in a single image.

We are unable to separate the indirect flash lighting from the ambient lighting, therefore our no-flash pixels capture the combined ambient plus indirect flash lighting.

We build on methods that combine a flash and no-flash image of a scene to produce a new image containing the desirable properties of both [ED04; PSA*04; ARNL05]. We recover a high resolution detail layer from the flash portions of the image and a large scale intensity layer from the no-flash regions. We demonstrate single exposure flash/no-flash and coarse depth map estimation as applications of our multiplexed flash illumination.



(a) (b)

Figure 2: *Our prototype(a) consists of a DSLR camera and a film camera modified to project a high-frequency pattern through its main lens. (b) A binary mask is used to block flash rays and produce a spatially varying pattern at the flash focus plane.*

3 Multiplexed Illumination

We divide the flash beam into a grid of pixels and allow each pixel to be either on or off. If a flash pixel is on, light is projected onto the scene and focused at the focal plane of the camera. If a flash pixel is off, light is blocked and does not enter the scene. Figure 2(b) shows a diagram of our optical system. We do not assume that the flash and camera are coaxial (i.e. no beam-splitter). We have found that the beam splitters necessary for coaxial illumination suffer from loss of light and flare. We only assume that flash and camera are loosely aligned.

3.1 Hardware Prototype

In order to achieve spatially varying flash intensities, we augment a traditional photographic flash with a binary mask pattern and focusing optics. In essence, we turn a traditional flash into a flash projector. The key distinction between our modified flash and a projector is that our flash produces a short burst of light as opposed to continously illuminating the scene, which is essential for freezing motion in photographs. While a projector can be used to simulate our flash, particularly for static scenes, we found there were a number of disadvantages to using a standard consumer projector. In particular, projectors often have low contrast, poor optics (e.g. high chromatic aberation and lens distortion), and a wide fixed aperture providing very shallow depth of field. In our design, we used printed binary transparency masks with a very high contrast ratio and the focusing optics of a high quality professional SLR camera lens with low chromatic aberation and full aperture control in order to control depth of field. An image of our system is shown in Figure 2(a). The film camera body on top has been transformed into our ”flash projector” by removing the back and placing our mask at the original film plane. A standard flash is mounted behind the ”film plane” with a diffuser separating the flash and mask. Essentially, the camera is being used in ”reverse” – light is shone from the original image plane out through the lens, producing a focused version of the mask onto the scene. An additional feature of this design is that if the focusing lens is thrown completely out of focus, the flash pattern is removed (via defocus blur) and the multiplexed flash is restored back to a traditional flash[1]. This allows the flash to operate in two modes: traditonal and multiplexed flash.

3.2 Illumination Patterns

In this section we consider several possible patterns for the flash illumination including uniform, poisson-disk, and striped. Once a type of pattern is chosen, the main parameter we explore is the ratio of flash and no-flash pixels in a particular sampling pattern. This ratio has two direct consequences: the sampling rate (in the Nyquist sense) of the reconstructed flash and no-flash images and the total amount of flash light in the scene. In general, the ratio should be chosen such that the resulting sampling rate matches the frequency content of each component. Unfortunately, the frequency content cannot be known a priori, and we are forced to make decisions based on some estimate of expected frequency content and how important it is for the specific application. In particular, we observe that flash/no-flash techniques rely more on the high frequencies of the flash component and on the low frequencies of the no-flash one.

The total number of ”on” flash pixels affects the total amount of light sent into the scene, but not the direct light received by a given illuminated point; it only increases the fraction of illuminated points. However, as the ratio of flash pixels increases, this introduces more indirect flash light, ”corrupting” the no-flash pixels.

Uniform A uniform checkerboard produces an equal number of flash and no-flash samples, uniformly distributed and regularly spaced. The ratio of flash to no-flash samples can be adjusted to produce regularly spaced samples with greater or fewer no-flash samples. However, although the flash mask contains regularly spaced samples, parallax between the flash and the camera distorts the spacing of samples when imaged at the camera. This distortion makes localizing the flash vs. no-flash samples on the camera sensor more difficult than with traditional assorted pixel schemes.

Stripes A stripe pattern can help localize the flash and no-flash samples if the optical centers of the flash and camera are carefully aligned. In particular, we can constrain the epipolar geometry such that vertical lines in the flash mask are projected to vertical lines in the camera. A disadvantage of this pattern is that it yields a non-uniform sampling between the vertical versus horizontal dimensions.

Poisson-disk As mentioned above, applications of flash/no-flash pairs usually take their high-frequency information from the flash component. As a consequence, we may choose to undersample the no-flash component to increase the total flash intensity and record a larger number of well-exposed flash pixels. In order to hide some of the aliasing and noise that may occur, Poisson-disk distributed points can be used instead of a uniform grid when undersampling.

4 Reconstruction

Once we have captured a multiplexed flash illumination image, we must identify and separate the flash pixels from the no-flash pixels. Since we seek a direct simple extension of the traditional flash, the illumination and lens are not confocal and parallax makes it harder to identify which pixels are lit by the flash. Without geometric correspondence, we rely on statistical methods to determine

[1] with some loss in intensity due to the mask blocking light.

(a) (b) (c)

(d) (e) (f)

Figure 3: *Top row: A scene photographed with(a) and without(b) a standard flash. (c) Standard flash/no-flash image fusion. Our reconstructed flash(d) and no-flash(e) images and our single-exposure flash/no-flash reconstruction(f).*

flash and no-flash pixels. A simple method proposed by Nayar et. al. [NKGR06] is to choose flash pixels as the maximum pixels in some local window. Similarly, no-flash pixels are the minimum pixels in each local window. To reduce speckle noise, we compute a weighted average of the K largest and smallest pixels in a local window and use this as our estimate of flash and ambient pixels, respectively. The size of the window is chosen differently for flash and no-flash pixels and is based on the known ratio of flash to no-flash pixels.

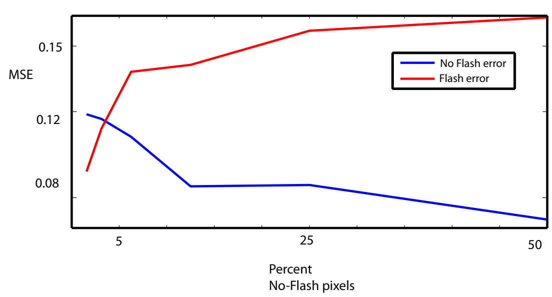


Figure 4: *Plot of reconstruction error as a function of the percentage of no-flash samples used in a uniform sampling pattern.*

Figure 4 shows a plot of the reconstruction error for the flash and no-flash components of our test scene (shown in Figure 3) as a function of the percentage of no-flash pixels in the flash pattern. Flash and no-flash images were taken separately and used as the ground truth. As expected, as the percentage of no-flash pixels increase, the no-flash reconstruction error decreases, and the flash reconstruction error increases. This graph suggests that there is little benefit to increasing the ratio of no-flash pixels above $\approx 20\%$. For our flash/no-flash application we use masks with $\approx 6 - 12\%$ no-flash pixels. This trade-off between capturing flash and no-flash pixels is similar to the spatial-angular tradeoff common to many lightfield camera designs [Ng05; GZC*06; GSMD07].

Improving Resolution As a consequence of using max and min operators to localize points, detail has a tendency to dilate or erode in the flash and no-flash images, depending on the local intensity gradient (see Figure 3(f) for an example). In order to improve sharpness and combat dilation and erosion in the flash image, we use texture synthesis to fill in missing data [EL99]. We remove a disk of pixels around each no-flash pixel location and infill these pixels with texture synthesis (see Figure 6). An additonal advantage of using Poisson-disk distributed no-flash samples is that the irregularity of the sampling hides artifacts that may occur when in-painting regions on a regular grid.

4.1 Improved localization

We have developed an algorithm to improve localization of flash and no-flash pixels when using a uniform grid illumination pattern. Because we do not coaxially align the flash projector and the camera there is parallax which makes localizing the no-flash pixels non-trivial. This is particularly evident across depth discontinuities and on highly curved surfaces. Depth discontinuites cause shifts in the stride between adjacent flash or no-flash pixels. Curved surfaces cause a row (or column) of points to be projected along a curve instead of along a straight line. However, locally (within a small neighborhood) the projected flash pattern is often very similar to a uniform grid. The general idea of our algorithm is to identify likely flash and no-flash pixels and then iteratively propagate local

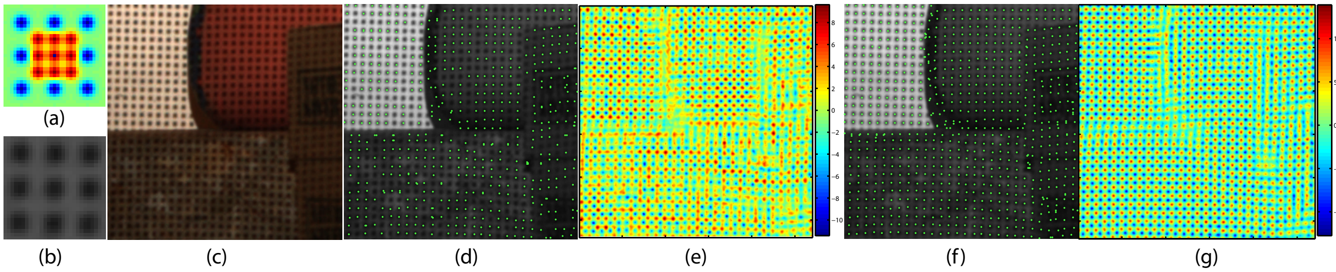


Figure 5: *Improving localization. (a) A typical ϕ_{min} kernel. (b) A close-up of the flash pattern projected on a scene. Notice that ϕ_{min} closely resembles (b). (c) An input scene. The inital estimate of no-flash pixel locations (d) and the corresponding P map (e). Notice that (d) has many missing pixels locations and is lacking structure. (f) shows the final estimate of no-flash pixel locations and the final P map (g) after 20 iterations. Our localization method is able to propagate local structure and accurately identifies no-flash pixels.*

evidence to influence the estimate of nearby locations.

Initialization We initialize the estimated locations using a method similar to Nayar et. al. [NKGR06], finding the maximum or minimum pixels in non-overlapping $M \times M$ windows, where M is chosen to match the projected size (or stride) of the illumination pattern in camera pixels. We note that if the focal lengths of the flash projector and the camera are matched then the size of the projected pattern (magnification) is not affected by scene depth or parallax.

Propagating local evidence Given an initial estimate of the flash and no-flash pixel locations, F and NF respectively, we wish to refine them by incorporating a local spatial model of the relative positions between adjacent flash pixels. The intuition is that if we have found the location of one flash pixel, we can use this information to help estimate the location of neighboring flash pixels.

To propagate information we construct a map P as:

$$P = F * \phi_{max} + NF * \phi_{min}. \quad (1)$$

where F and NF are indicator images that have, e.g. $NF(p) = 1$ for no-flash pixels p and zero otherwise, ϕ_{max} and ϕ_{min} are kernels that encode the relative spatial locations of other flash pixel locations as signed functions and $*$ denotes convolution. For example, ϕ_{max} is positive where we expect to find flash pixels, negative where we expect to find no-flash pixels and zero otherwise. We set $\phi_{min} = -\phi_{max}$.

We iteratively perform a sequence of steps designed to find pixel locations that simultaneously agree with the input data (e.g. are local maximums or minimums) and are appropriately spaced relative to neighboring flash and no-flash pixels. First, we build

$$P_{max} = P \times I_{gray} \quad (2)$$
$$P_{min} = P \times (1 - I_{gray}) \quad (3)$$

where I is a grayscale ([0-1] normalized) version of the input image I. P_{max} and P_{min} reweight P, giving more weight to flash pixels locations that are in bright parts of the image, and more weight to no-flash pixels locations in dark parts of the image. As P_{max} and P_{min} are processed symmetrically - MAX can be substituted for MIN (and vice-versa)- the remaining steps will be described for computing P_{min} only. We find the set Q of local maxima of the laplacian $\nabla^2 P_{min}$ with response greater than a threshold τ:

$$Q = \left\{ q \,\middle|\, \nabla^2 P_{min}(q) > \tau \wedge q = \arg\max_{p \in \Omega_q} \nabla^2 P_{min}(p) \right\}. \quad (4)$$

In practice we use a local window of 5×5 pixels, and a threshold $\tau = 2$. Local maxima of $\nabla^2 P_{min}$ are points where the gradient is increasing quickly (e.g. at the minimum of no-flash pixels) and we threshold to discard points with small response. We use Q to update our current estimate of no-flash pixel locations NF as:

$$\forall q \in Q, NF(q) = CLAMP(\nabla^2 P_{min}(q) - R_0)/R_1, 0, 1) \quad (5)$$

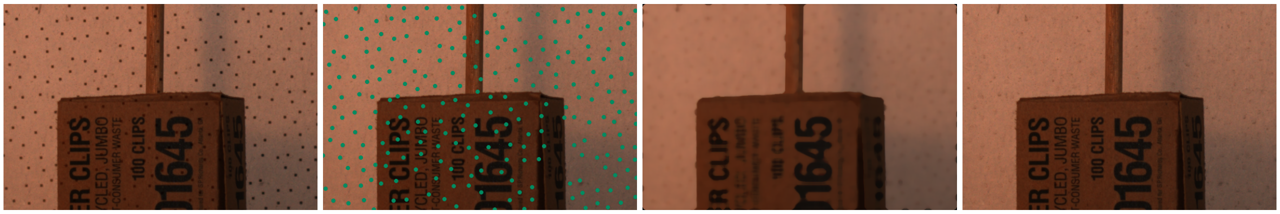
which linearly maps the range [R_0,R_1] to [0,1] and clamps values outside the range (we found [R_0,R_1] = [1,5] to work well in practice). We set $NF(p) = 0$ for all $p \notin Q$. Finally, we recalculate P (using Equation 1) and iterate. After K iterations we calculate the final flash and no-flash pixel positions by thresholding F and NF. In practice we run $K = 20$ iterations and use a threshold of 0.2. Figure 5 shows an example of P and NF before and after running our iterative estimation algorithm.

5 Depth from flash defocus

Similar to [MNBN07], we can use the flash projector defocus to estimate a coarse depth map of the scene. However, there are several distinctions between our work and previous approaches. First, we do not assume the flash projector and the camera are coaxially aligned, which introduces parallax and makes the localization more challenging. We describe a method to improve localization in Section 4.1. A second fundamental difference between our setup and the one described by Moreno-Noguer and colleagues is that we aim for an infinite contrast ratio[2] between flash and no-flash pixels while they specifically illuminate the entire scene with some baseline illumination. We aim for an infinite contrast ratio because we wish to recover only no-flash illumination in the no-flash pixels. One advantage of Moreno-Noguer and colleagues approach[MNBN07] is that they are able estimate and "invert" the projector illumination blur because it is nonzero everywhere. Our goal is to estimate a sparse depth map by analyzing the blur at each no-flash pixel, and we rely on the previously mentioned methods to improve the resolution of the flash image (Section 4).

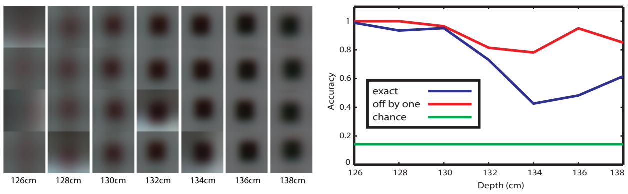
Patch Database Our approach is to construct a database D of examplar patches e_d that model how flash defocus changes as a function of scene depth d. In order to build our database we take multiple photographs a planar scene containing patches with different albedos over a range of depths. The camera and flash focus remain fixed for all images, as the distance d to the planar scene is varied from d_{min} to d_{max} producing a stack of images $\{I_d\}$. We used a relatively small aperture for the camera ($f/10$) and a large aperture for the flash projector ($f/2.8$) to ensure that most of the observed defocus is due to the flash and not the camera. From

[2] In practice this is impossible - due to indirect illumination, defocus, and the finite contrast of the occluding mask.



(a) (b) (c) (d)

Figure 6: *Using texture synthesis to improve resolution. (a) multiplexed illumination image. (b) No-flash pixels labeled and disk of pixels around each is marked. Standard reconstruction(c) dilates and blurs features. Texture synthesis fills in missing points and avoids resolution loss.*



(a) (b)

Figure 7: *a) A patch database for seven depths ranging from 126cm to 138cm in 2cm increments. Each depth has $K = 4$ exemplar patches. b) Error plot testing our depth estimation method. The blue curve shows the percentage points assigned the correct depth label as a function of depth. The red curve shows the percentage of points assigned the correct depth label, or a label ± 1 from the correct label. In this case a mislabeling by 1 corresponds to a 2cm error in depth estimation. The green curve shows the performance of assigning depth labels at random.*

each image I_d we estimate the no-flash pixel locations and crop a $N \times N$ window around each no-flash pixel creating a large collection of example patches for each depth. We use k-means clustering to compute K examplar patches $e_d^k, k = 1 \dots K$ for each depth d, and the set of all these examplars over all depths forms our database $D = \left\{ e_d^k | k = 1 \dots K, d \in [d_{min}, d_{max}] \right\}$. In order to provide albedo invariance, we independantly normalize each color channel of e_d^k to have unit mean. Figure 7 shows a database of patches for 7 depth values ranging in 2cm increments from 126cm to 138cm. For each depth d we have computed $K = 4$ exemplar patches.

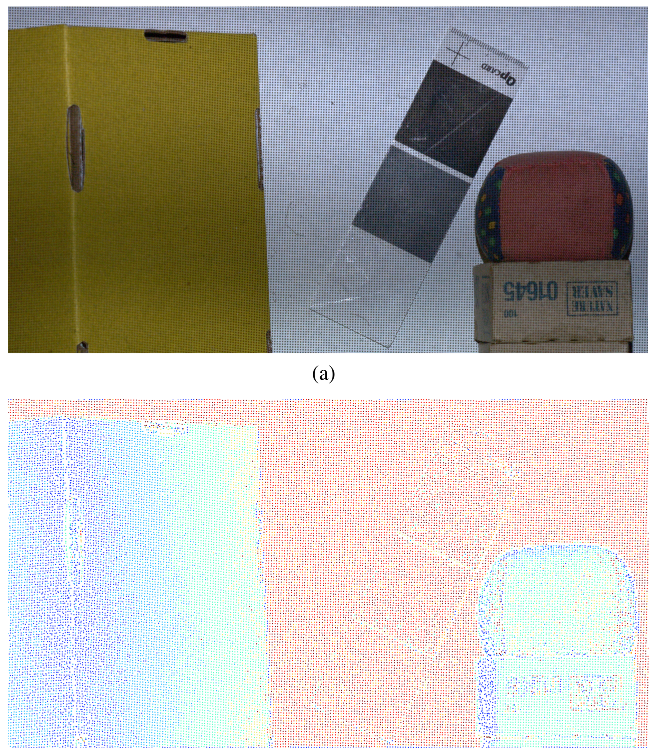
Estimating Depth Given a new scene, we would like to estimate depth at each no-flash location p. Let l_p be the $N \times N$ window of pixels centered at p, and $\hat{\mu_p}$ be the per-color channel (i.e. RGB) mean of l_p. We compute the error $E(l_p, d)$ for depth d as:

$$E(l_p, d) = \min_{k=1 \dots K} \left\| l_p - \hat{\mu_p} \cdot e_d^k \right\|^2 . \tag{6}$$

We rescale the examplar patches e_d^k by the RGB means $\hat{\mu_p}$, instead of normalizing l_p to unit means per channel in order to avoid amplifying noise in l_p. For example a blue object may have a very low red channel, and thus normalizing the red channel to unit mean would amplify any noise present. Conversely, weighting e_d^k by $\hat{\mu_p}$ will downweight the importance of the red channel when computing the error. In the simplest case we use nearest neighbor classification and select the d^* that minimizes $E(l_p, d)$ as the depth at pixel p:

$$d^* = \arg\min_d E(l_p, d) \tag{7}$$

We can add spatial regularization using a markov random field (e.g. graph cuts) [BVZ01].



(a)

(b)

Figure 8: *a) Multiplexed flash illumination input image. b) Sparse depth map computed at each no-flash pixel. Blue values are closer to the camera. Red values are further away.*

Figure 7 shows an error plot of the number of correctly classified points as a function of depth, for a set of seven images of a planar scene, covering the depth range 126cm to 138cm in 2cm increments, using nearest neighbor classification. The seven test images were the same images used to create the patch database. Each test image contained approximately 8300 no-flash pixels. The y-axis of the plot shows the percentage of points correctly label as a function of depth. On the low end, points at 134cm were correctly identified 42% of the time, whereas on the high end, points at 126cm were correctly identified 98% of the time. Chance would correctly label points 14% of the time. In addition, the curve marked ”off by one” shows the percentage of points that were assigned a depth label off by at most one from the correct label (corresponding to a depth error of 2cm in our experiment). This improves the percentage to greater than 78% of points.

Figure 8 shows results for a scene with depth variation over the full working range. The yellow box on the left is slanted away and our depth map reflects this. Also note the bean bag and brown box are estimated at the same depth, as are the different segments of the gray card, disregarding the significant difference in albedos.

6 Single exposure flash / no-flash

To demonstrate our multiplexed illumination, we show single exposure flash/no-flash on a dynamic scene. Traditional flash /no-flash methods[ED04; PSA*04] take as input a flash and a no-flash image of the same scene. These methods assume there is minimal motion between flash and no-flash images (such that a simple alignment will produce pixel level correspondence). Next, the images are decomposed into detail and large-scale layers using the bilateral filter (and other variants such as the cross/joint bilateral filter [ED04; PSA*04]). Finally a new image is synthesized by combining the detail layer of the well exposed, low noise flash image with the large-scale intensity layer of the under exposed and noisy flash image. In essence, this combines the sharp details of the flash image with the pleasing ambient lighting of the no-flash image.

Scenes with motion pose a problem for traditional flash/no-flash methods because it is no longer possible to align objects between exposures. Using our flash design, we are able to capture enough information in a single image to perform a flash/no-flash image fusion. Figure 9(a) shows a person tossing a bean bag, captured using a standard flash in order to freeze the motion of the object. Figure 9(b) shows a no-flash image taken of the same scene shortly aftwards. Objects have changed position in the no-flash image, and there is a large amount of motion blur. Figure 9(c) shows the results of performing flash/no-flash fusion using the components captured from a single image. In this example, we used Poisson-disk distributed no-flash points and reconstructed the flash image using texture synthesis to fill in missing data. Our result has the sharpness of the flash image, as well as the shadowing and glow of the no-flash image.

7 Discussion

Flash multiplexing shows promise for computational illumination in dynamic scenes because it enables the simultaneous capture of multiple components of illumination. Our prototype is able to multiplex flash and ambient lighting into assorted flash pixels captured at the image sensor. The defocus of the light pattern further allows us to extract simple depth information. As an application of our multiplexed flash illumination, we demonstrate the first single-exposure flash/no-flash method suitable for dynamic scenes.

Illumination multiplexing raises challenging issues. A limitation of our method is the assumption that no-flash pixels capture only ambient lighting. In practice, these pixels are illuminated not only by the ambient lighting, but also by the indirect light from the flash. Additionally, there will be some light spill due to defocus of nearby flash pixels and the finite contrast of the transparency mask used to create our sampling pattern. We want to explore ways to further separate the recovered no-flash image into true ambient and flash indirect lighting. Recent work on multi-light white balance [HMP*08] may help accomplish this separation. Currently, we use texture synthesis to improve the resolution of the flash image. However texture synthesis is computationally expensive when running on large images, so we would like to explore other local methods to improve resolution. Finally, we would like to extend our method to work with video.

References

AGRAWAL A., RASKAR R., NAYAR S. K., LI Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM Press, pp. 828–835.

BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001).

DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Computer Graphics* (July 2000), SIGGRAPH 2000 Proceedings, pp. 145–156.

EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. In *ACM Transactions on Graphics (Proceedings of Siggraph Conference)* (2004), vol. 23, ACM Press.

EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *ICCV (2)* (1999), pp. 1033–1038.

GREEN P., SUN W., MATUSIK W., DURAND F.: Multi-aperture photography. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (2007). [to appear].

GEORGIEV T., ZHENG K. C., CURLESS B., SALESIN D., NAYAR S., INTWALA C.: Spatio-angular resolution tradeoffs in integral photography. In *Proceedings of Eurographics Symposium on Rendering (2006)* (June 2006), pp. 263–272.

HSU E., MERTENS T., PARIS S., AVIDAN S., DURAND F.: Light mixture estimation for spatially varying white balance. *ACM Trans. Graph. 27*, 3 (2008), 1–7.

LEVOY M., CHEN B., VAISH V., HOROWITZ M., MCDOWALL I., BOLAS M.: Synthetic aperture confocal imaging. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 825–834.

MORENO-NOGUER F., BELHUMEUR P. N., NAYAR S. K.: Active refocusing of images and videos. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM Press, p. 67.

MIAO X., SIM T.: Ambient image recovery and rendering from flash photographs. In *ICIP* (2005), pp. II: 1038–1041.

NG R.: Fourier slice photography. *ACM Transactions on Graphics 24*, 3 (Aug. 2005), 735–744.

NAYAR S., KRISHNAN G., GROSSBERG M. D., RASKAR R.: Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)* (Jul 2006).

NAYAR S., MITSUNAGA T.: High Dynamic Range Imaging: Spatially Varying Pixel Exposures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 2000), vol. 1, pp. 472–479.

NAYAR S., NARASIMHAN S.: Assorted Pixels: Multi-Sampled Imaging With Structural Models. In *Europian Conference on Computer Vision (ECCV)* (May 2002), vol. IV, pp. 636–652.

PETSCHNIGG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3 (2004), 664–672.

SEN P., CHEN B., GARG G., MARSCHNER S. R., HOROWITZ M., LEVOY M., LENSCH H. P.: Dual Photography. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005).*

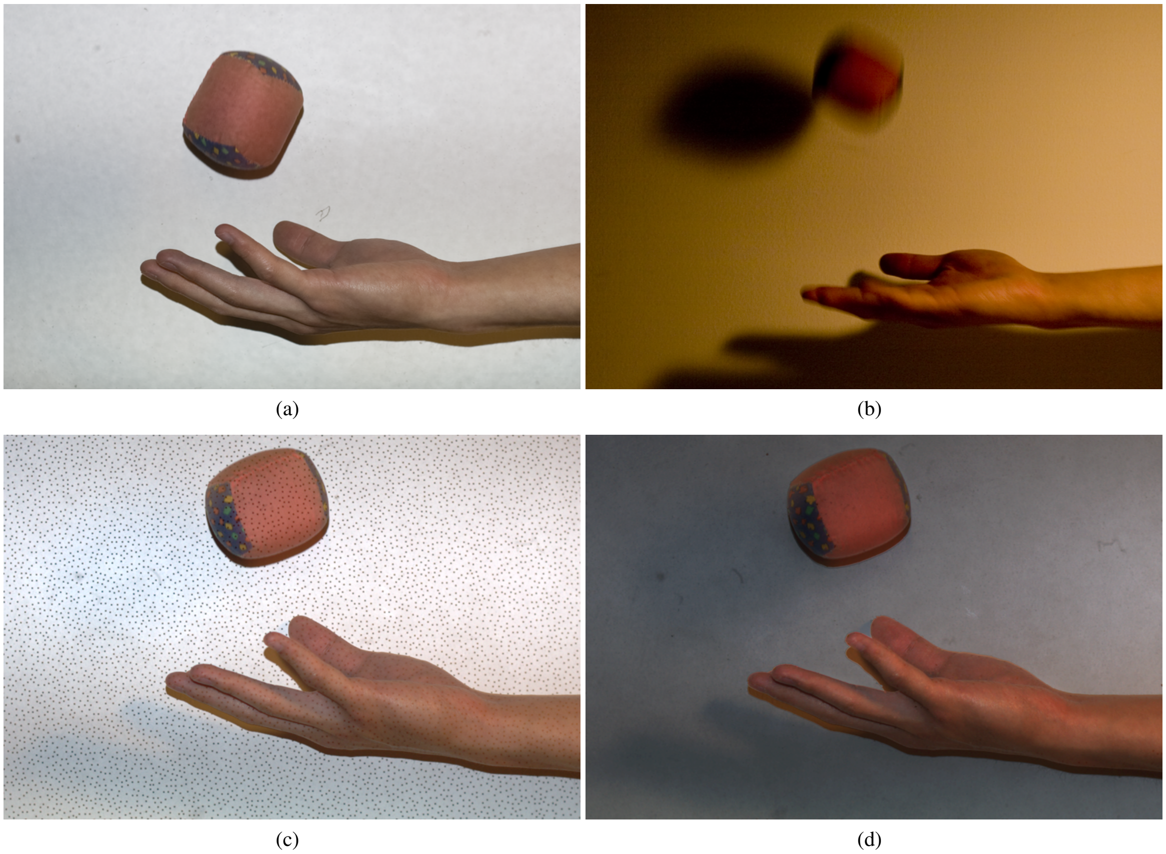


Figure 9: *The motion in a dynamic scene is frozen with standard flash phography(a) but the soft ambient light is lost. Two image flash cannot be used because the no-flash image(b) has changed and is blurry. From our multiplexed illumination image(c) we can create a new image that freezes the motion and retains the character of the ambient lighting.*

SCHECHNER Y. Y., NAYAR S. K., BELHUMEUR P. N.: Multiplexing for optimal lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 8 (2007), 1339–1354.

WENGER A., GARDNER A., TCHOU C., UNGER J., HAWKINS T., DEBEVEC P.: Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics 24*, 3 (July 2005), 756–764.

ZHANG L., NAYAR S. K.: Projection Defocus Analysis for Scene Capture and Image Display. *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)* (Jul 2006).